Limitations of Information-theoretic Incompressible Encodings

Petr Sedláček
Charles University, Faculty of Mathematics and Physics
sedlacek@karlin.mff.cuni.cz

Abstract

In this note we study the limitations of incompressible encodings with information-theoretic security. We demonstrate a flaw in the existing proof of the impossibility of constructing incompressible encodings information-theoretically. Our main contribution is a full proof of impossibility of existence of non-trivial information-theoretically secure incompressible encoding schemes.

Keywords

incompressible encodings, plain model, information-theoretic security

1 Introduction

The focus of this note is a primitive called *incompressible encoding scheme*, recently introduced by Moran and Wichs [MW20]. An incompressible encoding scheme gets some data as an input message and encodes it into an effectively incompressible code word. The original data can be easily decoded from the code word by anyone. However, it is not possible for any probabilistic polynomial-time adversary to compress the code word so that they can later decompress it to the code word, even knowing the original message.

In their article, Moran and Wichs provided constructions of incompressible encodings under various computational hardness assumptions as well as negative results regarding their security. They also studied various other properties of incompressible encodings, such as their composability.

1.1 Our contributions

Moran and Wichs conjectured that it is not possible to create a non-trivial information-theoretically secure incompressible encoding, i.e., without any computational restrictions on the adversary. In this note, we show that the sketch of a proof given in [MW20] has a few significant issues, which we describe in detail in Section 3. We present our own full proof of the impossibility of existence of information-theoretically secure incompressible encodings in the plain model in Section 4. (See Section 4 also for a high-level overview of our proof.)

Our results are somewhat analogous to the limitation of perfectly secure symmetric encryption. The one time pad scheme is secure against unbounded attackers, but the key must be as long as the message. A classical result by Shannon shows that this cannot be improved - every symmetric encryption scheme resilient to unbounded attackers cannot have the key shorter than the message. That gives solid ground to the studies of "weaker" encryption schemes that rely on computational hardness assumptions and are secure only against computationally bounded adversaries. Our full proof of the impossibility of non-trivial incompressible encoding scheme resilient against computationally unbounded adversaries substantiates focused research of incompressible encodings relying on computational hardness assumptions.

Figure 1: Compression experiment $\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda)$

For all encoding schemes $\Pi = (\mathrm{Enc}, \mathrm{Dec})$, and adversaries $\mathcal{A} = (\mathcal{A}.\mathrm{Select}(1^\lambda), \mathcal{A}.\mathrm{Compress}(\mathrm{aux}, c), \mathcal{A}.\mathrm{Expand}(\mathrm{aux}, w))$, and $\beta = \beta(\lambda, k)$, the compression experiment $\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda)$ is defined as follows:

$\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda)$

1. $(m, \mathrm{aux}) \leftarrow \mathcal{A}.\mathrm{Select}(1^\lambda)$.
2. $c \leftarrow \mathrm{Enc}(1^\lambda, m)$.
3. $w \leftarrow \mathcal{A}.\mathrm{Compress}(\mathrm{aux}, c)$.
4. $c' \leftarrow \mathcal{A}.\mathrm{Expand}(\mathrm{aux}, w)$.
5. Output 1 if and only if $c = c'$ and $|w| \leq \beta(\lambda, |m|)$.

2 Incompressible encodings

We use $\lambda \in \mathbb{N}$ to denote the security parameter.

Definition 2.1. An *Encoding scheme* $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ for a message space $M = \{m;\, m \in \{0,1\}^k\}$ consist of a pair of *Probabilistic Polynomial Time* (PPT) algorithms *Encode* ($\mathrm{Enc} : M \to \{0,1\}^*$) and *Decode* ($\mathrm{Dec} : \{0,1\}^* \to M$).

Definition 2.2 (p-correctness)**.** Let $p\colon \mathbb{N} \to [0,1]$ be a function. An encoding scheme $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ is *p-correct* if: $\forall \lambda \in \mathbb{N}, \forall m \in M, \Pr[\mathrm{Dec}(\mathrm{Enc}(1^\lambda, m)) = m] \geq p(\lambda)$.

Definition 2.3. We define a *negligible function* as a function $f : \mathbb{N} \to \mathbb{R}$ satisfying that for every $n \in \mathbb{N}$ there exists $K_n \in \mathbb{N}$ such that for every $\lambda \in \mathbb{N}$, $\lambda \geq K_n$ it holds that $\lambda^{-n} > |f(\lambda)|$. We denote the set of all negligible functions as $\mathrm{negl}(\lambda)$.

Definition 2.4 ((α,β)-incompressibility)**.** Let $\alpha, \beta : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be functions. Then encoding scheme $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ is (α,β)-*incompressible* if the following holds:

1. α-Bounding: $\forall m \in \{0,1\}^k, \forall \lambda \in \mathbb{N} : \Pr[|\,\mathrm{Enc}(1^\lambda, m)| \leq \alpha(\lambda, k)] = 1$
2. β-Incompressibility: For each adversary $\mathcal{A} = (\mathcal{A}.\mathrm{Select},\ \mathcal{A}.\mathrm{Compress},\ \mathcal{A}.\mathrm{Expand})$, it holds that $\Pr[\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda) = 1] \in \mathrm{negl}(\lambda)$, where the $\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}$ denotes a compression experiment defined in Figure 1 and visualized in Figure 2.

Consider the following construction. We can append a random string of length r to any message m of length k. The random part cannot be compressed and, thus, $\alpha(k) = k + r$ and $\beta(k) = r$. Therefore, encoding schemes satisfying $\beta \leq \alpha(k) - k$ are easy to obtain. From now on, we will focus solely on *non-trivial* encoding schemes satisfying $\beta(k) > \alpha(k) - k$.

Definition 2.5. A non-trivial (α,β)-incompressible encoding scheme is a p-correct (α,β)-incompressible encoding scheme with $\beta(\lambda, k) > \alpha(\lambda, k) - k$, and $(1 - p(\lambda)) \in \mathrm{negl}(\lambda)$.

In the rest of the note we rely on the following notation.

Notation 2.6.
For $k, \lambda \in \mathbb{N}$:

- $M := \{0,1\}^k$ is a set of all possible messages with length equal to k.

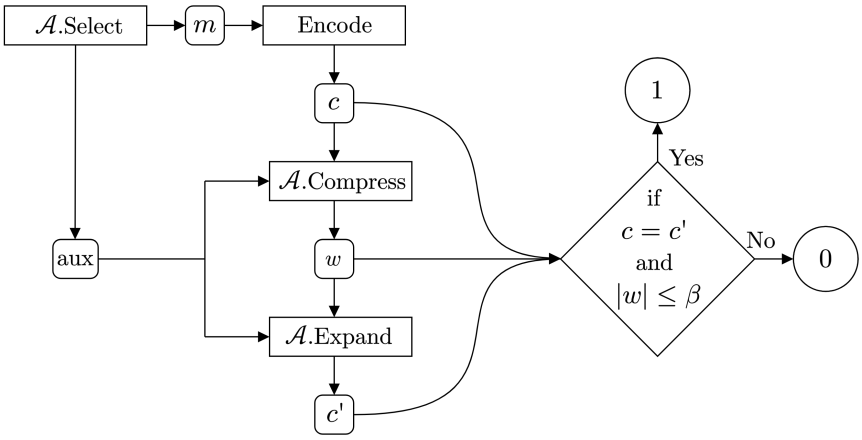

Figure 2: Compression experiment flow

- $W := \bigcup_{i=0}^{\alpha(\lambda,k)-k} \{0,1\}^i$ is a set of all bit strings with a length less than or equal to $\alpha(\lambda, k) - k$.
- $C := \{c \in \bigcup_{i=0}^{\alpha(\lambda,k)} \{0,1\}^i;\ \exists m \in \{0,1\}^k, \Pr[\mathrm{Enc}(1^\lambda, m) = c] > 0\}$ is the set of all possible code words for messages of a length k with respect to the security parameter λ.

For $m \in \{0,1\}^k$, $\lambda \in \mathbb{N}$:

- $C_m := \{c \in C;\ \Pr[\mathrm{Enc}(1^\lambda, m) = c] > 0\}$ is the set of all possible code words of the message m with respect to the security parameter λ.
- $D_m := \{c \in C; \mathrm{Dec}(c) = m\}$ is the set of all code words that are decoded to the message m.[1]

The use of the notation is depicted in Figure 3. The message m_1 can be encoded to any code word c from C_{m_1}, which is a subset of the set of all code words C. Every code word from the set D_{m_2} is decoded to the message m_2.

[1]Note that we use the notation D_m only when we assume the algorithm Dec being deterministic.

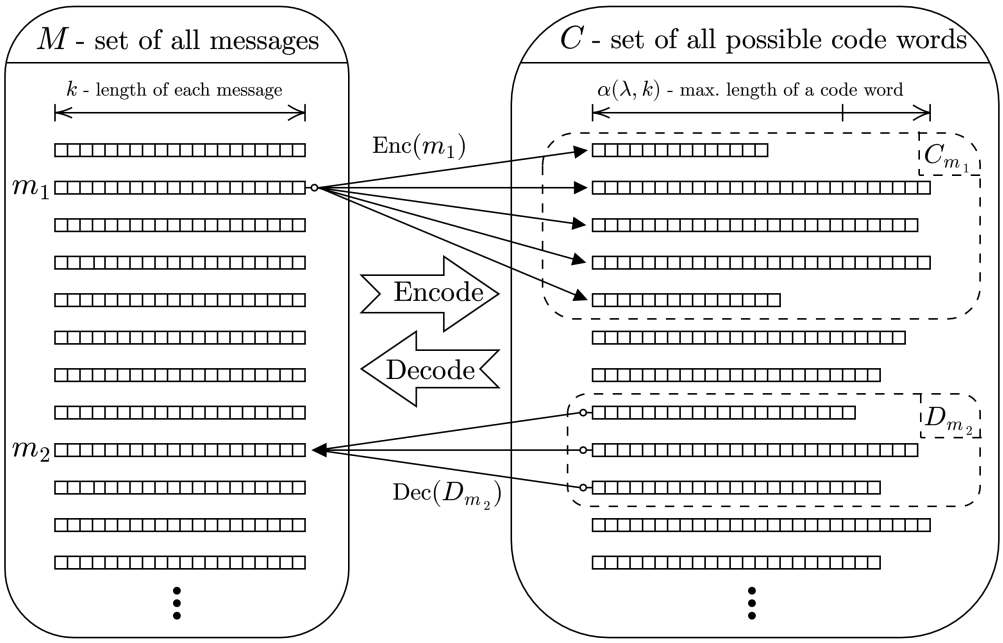

Figure 3: Encoding and decoding sketch

3 Issues in the proof-sketch in [MW20]

Statement 3.1 (Original sketch of proof by Moran and Wichs)**.** *"It is easy to see that non-trivial incompressible encodings cannot be constructed information theoretically. This is because, there are at most* $2^{\alpha(k)-k}$ *possible code words per message on average, and therefore also certainly for the worst-case message* m*. A pair of inefficient compression/ decompression procedures can enumerate the list of all such code words (e.g., in lexiographic order) and compress/decompress any code word in the list just by writing down its index using* $\beta(k) = \alpha(k) - k$ *bits."*

In their article, Moran and Wichs presented a broad definition of incompressible encodings and provided two possible constructions. The statement above holds for those constructions, but it does not hold for every construction possible that the definition allows. We identified two main issues.

The problematic part of the sketch is this sentence: *"This is because, there are at most* $2^{\alpha(k)-k}$ *possible code words per message on average, and therefore also certainly for the worst-case message* m*".* The first issue is that, in a probabilistic encoding (respectively decoding), each message can be encoded to (respectively decoded from) any number of code words with non-zero probability. Although to satisfy the *p-correctness*, some of those probabilities must be negligible. We provide a scheme that violates the above assumption from the proof sketch of [MW20] in the next section.

The second issue is that the notation of $\alpha(k)$ allows different lengths of code words. Therefore, there are more than $2^{\alpha(k)}$ possible code words. This issue is a technical one and can be solved in the following way. The adversary $\mathcal{A}$ is allowed to use the length of a bit sequence as an additional information, i.e., $\mathcal{A}$ is able to distinguish between "101" and "00101". Then the effects of the variable code word length cancel out (with slight technical difficulties). However, both of those issues show that the intuition presumed by [MW20] is unsubstantiated because there could be more than $2^{\alpha(k)-k}$ possible code words per message on average.

To summarise, the proof sketch works only for incompressible encoding schemes where the decoding is deterministic and p from the definition of p-correctness is equal to one.

3.1 Counterexample

Idea. We leverage an arbitrary encoding scheme and "weaken" its encoding algorithm in a way that allows every message to be encoded into any possible code word. Clearly, this scheme is of no practical interest but it is an explicit counterexample to the assumptions of [MW20]

Lemma 3.2. *For all p-correct* (α, β)*-incompressible encoding scheme* Π *there exists a* $\widehat{p}$*-correct* (α, β)*-incompressible encoding scheme* $\widehat{\Pi}$*, such that* $(p - \widehat{p}) \in \mathrm{negl}(\lambda)$ *and for all* $\lambda \in \mathbb{N}$*,* $m \in \{0,1\}^k$*,* $C_m = C$*, i.e., each message can be encoded into any possible code word with a non-zero probability.*

Proof. Let $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ be a p-correct, (α, β)-incompressible encoding scheme. For this example, we need an arbitrary efficiently computable negligible function, which we denote as $q(\lambda)$.

We define the encoding algorithm $\widehat{\mathrm{Enc}}$ in the following way. Before encoding a message m, $\widehat{\mathrm{Enc}}$ generates a uniformly random number $p \in [0,\, 1]$. Then it encodes the message m in the following way:

$$\widehat{\mathrm{Enc}}(m) = \begin{cases} \mathrm{Enc}(m) & \text{if } p \in (q(\lambda),\, 1], \\ \widehat{c}, & \text{if } p \in [0,\, q(\lambda)], \end{cases}$$

where $\widehat{c}$ is a uniformly chosen random code word satisfying $|\widehat{c}| \leq \alpha(\lambda,\, k)$ generated "on the fly".

$\widehat{\mathrm{Enc}}(m)$ is correctly defined, because $\forall \lambda > 1,\ q(\lambda) \in [0,\, 1)$. Next, we discuss the properties of $\widehat{\Pi} = (\widehat{\mathrm{Enc}},\, \mathrm{Dec})$.

First, we verify that $\widehat{\Pi}$ is indeed an $(\alpha,\, \beta)$-incompressible encoding scheme. The α-bounding property holds by the definition of $\widehat{\mathrm{Enc}}$. Next, we focus on β-incompressibility. We know from the definition of Π that an arbitrary adversary $\mathcal{A}$ wins the compression experiment against the Enc algorithm with at most negligible probability. In the $\widehat{\Pi}$ scheme, the probability of the message being encoded as a randomly chosen code word is negligible. In all other cases, the message is encoded via the Enc algorithm. Therefore, the chance of success of the adversary $\mathcal{A}$ against the $\widehat{\mathrm{Enc}}$ algorithm is also negligible.

The $\widehat{\Pi}$ scheme is $\widehat{p}$-correct, where $\widehat{p}(\lambda) \geq (1 - q(\lambda))p(\lambda) = p(\lambda) - p(\lambda)q(\lambda)$ because the Dec algorithm correctly decodes a code word encoded by Enc with the probability $p(\lambda)$ and the Enc algorithm is used for encoding with the probability $(1 - q(\lambda))$. It follows from the definition of negligible function that $p(\lambda)q(\lambda)$ is negligible. Thus, the $\widehat{\Pi}$ scheme satisfies the definition of a $\widehat{p}$-correct $(\alpha,\, \beta)$-incompressible encoding scheme.

Let us denote C the set of all possible code words. From the definition of $\widehat{\mathrm{Enc}}$ follows that

$$\forall c \in C, \forall m \in M :\ \Pr[\widehat{\mathrm{Enc}}(m) = c] \geq \frac{q(\lambda)}{|C|}.$$

That means that every message could be encoded to any possible code word with a non-zero probability, thus the proof is complete. □

4 Impossibility of information-theoretic security

Theorem 4.1. *Let* Π *be a p-correct,* α*-bounded encoding scheme. Then*

$$\forall \beta(\lambda, k) > \alpha(\lambda, k) - k\ \exists \mathcal{A} : \Pr[\mathrm{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^{\lambda}) = 1] \geq p(\lambda).$$

If an α-bounded encoding scheme Π satisfies that $(1 - p(\lambda)) \in \mathrm{negl}(\lambda)$, then from Theorem 4.1 it follows that $\beta(\lambda, k) \leq \alpha(\lambda, k) - k$. Hence, Π does not satisfy the definition of non-trivial incompressible encoding scheme. Thus, the following corollary holds.

Corollary 4.2. *It is not possible to construct a* non-trivial incompressible encoding scheme *information-theoretically.*

In the followings sections, we prove Theorem 4.1. For an easier understanding of the proof, we start with some additional assumptions on the encoding scheme. First, we assume perfect correctness and deterministic decoding. It allows us to leverage the idea from [MW20] because their proof sketch works for restricted schemes where the p from the definition of p-correctness is equal to 1 and the decoding is deterministic.

The core idea behind our proof for general incompressible encodings is the following. We look for a subset S of the set of all code words that satisfies two conditions. The first is that there exists a message m^*, such that the message is encoded into this subset S with probability at least p. The second requirement is that the subset is small enough, i.e., $|S| < 2^{\alpha(k)-k+1}$, so that it can be indexed using at most $\alpha(k) - k$ bits. Using the subset S, we create an adversary that is able to break the scheme with probability at least p. The adversary lets the challenger encode the message m^*. With probability at least p, the resulting code word is an element of the subset S. Then the adversary compresses the code word by using only its index in S, which is at most $\alpha(k) - k$ bits long. Thus, the adversary succeeds in the compression experiment CompExp with probability at least p, which proves Theorem 4.1. Showing that a subset satisfying the above conditions must exist in every encoding scheme of our interest is the technical part of the generalised proof in Section 4.3.

4.1 Schemes with perfect correctness and deterministic decoding

Definition 4.3. An encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ has *perfect correctness*, if it is *p-correct* scheme with $p(\lambda) = 1\ \forall \lambda \in \mathbb{N}$.

Definition 4.4. An encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ has *deterministic decoding* if the algorithm Dec is deterministic, i.e., for each code word $c \in C$ there exists a unique message $m \in M$ such that $\text{Dec}(c) = m$ with probability one.

In Lemma 4.5, we show that when the decoding is deterministic there exists a message m such that the set of code words D_m is small enough that all its members can be indexed using at most $\alpha(\lambda,\, k) - k$ bits.

Lemma 4.5. *Let $\Pi = (\text{Enc}, \text{Dec})$ be an encoding scheme with deterministic decoding and message space $M = \{0,1\}^k$. If $|C| \leq 2^{\alpha(\lambda,k)+1} - 1$, then $\min_{m\in M} |D_m| \leq 2^{\alpha(\lambda,k)-k+1} - 1$.*

Proof. Suppose to the contrary that $\min_{m\in M} |D_m| > 2^{\alpha(\lambda,k)-k+1}-1$. Therefore, $\min_{m\in M} |D_m| \geq 2^{\alpha(\lambda,k)-k+1}$. Subsequently,

$$|C| = \sum_{m\in M} |D_m| \geq \sum_{m\in M} \min_{m\in M} |D_m| = 2^k \min_{m\in M} |D_m| \geq 2^k \left(2^{\alpha(\lambda,k)-k+1}\right) > 2^{\alpha(\lambda,k)+1} - 1,$$

where the first equality follows from Dec being deterministic and the second equality follows from $|M| = 2^k$. We derived that $|C| > 2^{\alpha(\lambda,k)+1} - 1$, a contradiction to the assumption about the cardinality of C. Hence, the Lemma holds. □

Using Lemma 4.5, we construct an adversary that wins the compression experiment CompExp using the index of a code word c in D_m as a form of compression.

Theorem 4.6. *Let $\Pi = (\text{Enc}, \text{Dec})$ be an α-bounded encoding scheme with deterministic decoding and perfect correctness. Then*

$$\forall \beta(\lambda,k) > \alpha(\lambda,k) - k\ \exists\, \mathcal{A} : \Pr[\text{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda) = 1] = 1,$$

i.e., it is not a non-trivial (α,β)-incompressible encoding scheme.

Proof. Our construction of the adversary $\mathcal{A} = (\mathcal{A}.\text{Select}, \mathcal{A}.\text{Compress}, \mathcal{A}.\text{Expand})$ is given in Figure 4. The $\mathcal{A}$.Select algorithm chooses a message m such that every member of D_m can be uniquely represented using an index (bit string) with a length less than or equal to β bits. Then the algorithm chooses an injective function f from D_m to the set of indices. The $\mathcal{A}$.Compress gets a code word c from the Enc algorithm and returns its index $w = f(c)$. Finally, the $\mathcal{A}$.Expand returns $f^{-1}(w)$, which is equal to c.

Figure 4: Adversary $\mathcal{A}$

Adversary $\mathcal{A} = (\mathcal{A}.\text{Select}(1^\lambda), \mathcal{A}.\text{Compress}(\text{aux}, c), \mathcal{A}.\text{Expand}(\text{aux}, w))$

$\mathcal{A}.\text{Select}(1^\lambda)$:

- Choose $m^* = \text{argmin}_{m \in M} |D_m|$.
- Choose an injective $f : D_{m^*} \to W$.
- Compute $f^{-1} : f(D_{m*}) \to D_{m^*}$ from f.
- Output $(m^*, \text{aux} = (f, f^{-1}))$.

$\mathcal{A}.\text{Compress}(\text{aux}, c)$:

- Parse aux as $\text{aux} = (f, f^{-1})$.
- Output $w = f(c)$

$\mathcal{A}.\text{Expand}(\text{aux}, w)$:

- Parse aux as $\text{aux} = (f, f^{-1})$.
- Output $c' = f^{-1}(w)$

All messages are k bits long, therefore $|M| = 2^k$. On the other hand, the code words are α bits long *at most*. Hence, all code word lengths from 1 to α bits are possible. The set of all possible code words is $C \subseteq \bigcup_{i=0}^{\alpha(\lambda,k)} \{0,1\}^i$. Thus, $|C| \leq \left|\bigcup_{i=0}^{\alpha(\lambda,k)} \{0,1\}^i\right| = 2^0 + 2^1 + \cdots + 2^{\alpha(\lambda,k)} = 2^{\alpha(\lambda,k)+1} - 1$. By perfect correctness of the scheme, for all $c \in C$ there exists a unique $m \in \{0,1\}^k$ such that $c \in D_m$. Equivalently, $C = \bigcup_{m \in M} D_m$ while for all m_i, $m_j \in M$ satisfying $(i \neq j)$ it holds that $D_i \cap D_j = \emptyset$.

Let $W = \bigcup_{i=0}^{\alpha(\lambda,k)-k} \{0,1\}^i$ be the set of bit strings defined in Notation 2.6. Then $|W| = 2^{\alpha(\lambda,k)-k+1} - 1$. We choose $m^* = \text{argmin}_{m \in M} |D_m|$ and let $f : D_{m^*} \to W$ be an injective function satisfying that for all $c_1, c_2 \in D_{m^*}$ such that $(c_1 \neq c_2)$ it holds that $f(c_1) \neq f(c_2)$. This function is possible to construct, because we have $|D_{m^*}| \leq |W|$ from Lemma 4.5. Consequently, we construct $f^{-1} : f(D_{m^*}) \to D_{m^*}$, such that $f^{-1}(f(c)) = c$. Then, the $\text{CompExp}^{\Pi}_{\mathcal{A},\beta}$ follows:

- $\big(m^*, \text{aux} = (f, f^{-1})\big) \leftarrow \mathcal{A}.\text{Select}(1^\lambda)$
- $c \leftarrow \text{Enc}(1^\lambda, m^*)$
- $w \leftarrow \mathcal{A}.\text{Compress}(\text{aux}, c)$
- $c' = \mathcal{A}.\text{Expand}(\text{aux}, w)$
- if $(c' = c) \wedge |w| \leq \beta(\lambda, k)$ output 1, else output 0

The $|w| \leq \beta(\lambda, k)$ holds, because $w \in W = \bigcup_{i=0}^{\alpha(\lambda,k)-k} \{0,1\}^i$. The $c = c'$ follows from f being injective, which implies that the inverse f^{-1} is defined for all $f(c) \in W$ and $f^{-1}\big(f(c)\big) = c$. Therefore, the compression experiment always outputs one. □

4.2 Schemes with imperfect correctness and deterministic decoding

In this section, we consider incompressible encodings schemes with imperfect correctness, i.e., where decoding succedes only with some probability $p(\lambda) < 1$. We show that the adversary $\mathcal{A}$ defined in Figure 4 wins the compression experiment with probability equal to or greater than the correctness $p(\lambda)$ of the scheme.

Lemma 4.7. *Let Π be a p-correct, α-bounded encoding scheme with deterministic decoding. Then*

$$\forall \beta(\lambda, k) > \alpha(\lambda, k) - k,\ \exists \mathcal{A}\colon \Pr[\text{CompExp}^{\Pi}_{\mathcal{A},\beta}(1^\lambda) = 1] \geq p(\lambda),$$

where $\mathcal{A}$ is the adversary defined in the Figure 4.

Proof. Recall $\mathcal{A}$ from Figure 4. $\mathcal{A}$.Select chooses the message $m^* = \text{argmin}_{m \in M} |D_m|$. The algorithm Enc outputs a code word $c_{m^*} \in C_{m^*}$. If $c_{m^*} \in D_{m^*}$, $\mathcal{A}$.Compress will output the $w = f(c_{m^*})$ and, consequently, $\mathcal{A}$.Expand will expand w to the original code word c_{m^*}. Therefore, if $c_{m^*} \in D_{m^*}$, then the adversary $\mathcal{A}$ will correctly compress and expand the code word c_{m^*}. Out of the definition of p-correctness follows $\Pr[c_{m^*} \in D_{m^*}] \geq p(\lambda)$. Hence, we get $\Pr[\text{CompExp}^{\Pi}_{\mathcal{A},\beta}(m^*) = 1] \geq p(\lambda)$. □

4.3 Schemes with imperfect correctness and probabilistic decoding

In this section, we present the complete proof of the Theorem 4.1 without any additional assumptions.

Because of the decoding being probabilistic, we cannot use the argument $\min_{m \in M} |D_m| \leq 2^{\alpha(\lambda,k)-k+1} - 1$ from Lemma 4.5. Instead, we show that there exists a message m^* and a set $S \subset C$, such that the S can be indexed using $\alpha(\lambda, k) - k$ bits at most and the probability of $\text{Enc}(m^*) \in S$ is greater than or equal to the correctness $p(\lambda)$ of the scheme. Using Lemma 4.9 we construct an adversary that succeeds when the message m^* is encoded into any code word in S, therefore $\Pr[\text{CompExp}^{\Pi}_{\mathcal{A},\beta}(m^*) = 1] \geq p(\lambda)$.

Notation 4.8. We denote the $p_{\text{ENC}}(m, c) = \Pr[\text{ENC}(m) = c]$ and the $p_{\text{DEC}}(c, m) = \Pr[\text{DEC}(c) = m]$.

Lemma 4.9. *Let $m^* \in \text{argmin}_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*)$, then there exists a set $S \subset C$ such that $|S| = 2^{\alpha(\lambda,k)-k+1} - 1$ and $\sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$.*

The proof of Lemma 4.9 uses the two following auxiliary lemmata.

Lemma 4.10. *There exists $m \in M$ such that $\sum_{c \in C} p_{\text{DEC}}(c, m) \leq 2^{\alpha(\lambda,k)-k+1} - 1$.*

Proof. The $|C| = 2^{\alpha(\lambda,k)+1} - 1$ and the $|M| = 2^k$. We suppose for the contrary, that

$$\forall m \in M : \sum_{c \in C} p_{\text{DEC}}(c, m) > 2^{\alpha(\lambda,k)-k+1} - 1,$$

that implies that

$$\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m) > 2^{\alpha(\lambda,k)-k+1} - 1.$$

Thence $\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m) \geq 2^{\alpha(\lambda,k)-k+1}$.

$$\begin{aligned}
|C| &= \sum_{c \in C} 1 = \sum_{c \in C} \left(\sum_{m \in M} p_{\text{DEC}}(c, m) \right) = \sum_{m \in M} \left(\sum_{c \in C} p_{\text{DEC}}(c, m) \right) \\
&\geq \sum_{m \in M} \left(\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m) \right) = |M| \min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m) \\
&\geq 2^k 2^{\alpha(\lambda,k)-k+1} = 2^{\alpha(\lambda,k)+1} \\
&> |C|.
\end{aligned}$$

We have a contradiction, thence the lemma holds. □

Lemma 4.11. *Let $n \in \mathbb{N}$, $K \in [0,\ n]$, $\mathbf{y} = (y_1,\ \ldots,\ y_n)$, such that for all $i, j \in \{1,\ \ldots,\ n\}$ $y_i \geq 0$ and $(i > j) \Rightarrow (y_i \leq y_j)$. $\mathcal{X} = \{\mathbf{x} = (x_1,\ \ldots,\ x_n) \mid \forall k \in \{1,\ \ldots,\ n\}\ x_k \in [0, 1],\ \sum_{k=1}^{n} x_k = K\}$.*

Let us define $\mathbf{z} = \begin{cases} 1 & \text{if } i \leq \lfloor K \rfloor, \\ K - \lfloor K \rfloor & \text{if } \lfloor K \rfloor < i \leq \lfloor K \rfloor + 1, \\ 0 & \text{if } i > \lfloor K \rfloor + 1. \end{cases}$

Then

$$\forall x \in \mathcal{X} : \sum_{k=1}^{n} x_i y_i \leq \sum_{k=1}^{n} z_i y_i,$$

i.e.

$$\mathbf{z} \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}} \sum_{k=1}^{n} x_i y_i.$$

Proof. First, we verify that $\mathbf{z} \in \mathcal{X}$.

- $z_k \in [0,\ 1]$ for all $k \in \{1,\ \ldots,\ n\}$, because $1 \in [0,\ 1], 0 \in [0,\ 1]$ and $K - \lfloor K \rfloor \in [0,\ 1]$.
- $\sum_{k=1}^{n} z_k = \underbrace{1 + \cdots + 1}_{\lfloor K \rfloor - times} + (K - \lfloor K \rfloor) = \lfloor K \rfloor + (K - \lfloor K \rfloor) = K.$

Let us suppose to the contrary that there exists $\bar{\mathbf{z}} \in \mathcal{X}$, such that

$$\bar{\mathbf{z}} \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}} \sum_{k=1}^{n} x_k y_k,\ \sum_{k=1}^{n} \overline{z_k} y_k > \sum_{k=1}^{n} z_k y_k.$$

We denote $i = \operatorname{argmin}_{k\in\{1,\cdots,n\}} z_i \neq \overline{z_i}$. We distinguish the following cases:

- If $i \leq \lfloor K \rfloor$. Then $z_i = 1$, $\overline{z_i} \in [0,\ 1]$ and $\overline{z_i} \neq z_i$. That implies $\overline{z_i} < z_i$.
- If $\lfloor K \rfloor < i \leq \lfloor K \rfloor + 1$. Then $z_i = K - \lfloor K \rfloor$. From the definiton of i we get $\overline{z_k} = z_k \forall k \in \mathbb{N},\ k < i$. Thus $\sum_{k=1}^{i-1} \overline{z_k} = \sum_{k=1}^{i-1} z_k = \lfloor K \rfloor$ and we know that $\sum_{k=1}^{n} \overline{z_k} = K$. That implies that $\overline{z_i} \leq K - \lfloor K \rfloor = z_i$. Then $\overline{z_i} \neq z_i$ implies $\overline{z_i} < z_i$.
- If $i > \lfloor K \rfloor + 1$, then from the definition of i we obtain that for all $k \in \mathbb{N},\ k < i\colon\ \overline{z_k} = z_k$. That implies $\sum_{k=1}^{i-1} \overline{z_k} = \sum_{k=1}^{i-1} z_k = K$. However, $\bar{\mathbf{z}} \in \mathcal{X}$. Thus for all $k \geq i : \overline{z_k} = 0 = z_k$, which contradicts the definition of i. Hence $i \leq \lfloor K \rfloor + 1$.

We notice that $\sum_{k=1}^{i} \overline{z_k} < \sum_{k=1}^{i} z_k \leq K$ and $\bar{\mathbf{z}} \in \mathcal{X} \Rightarrow \sum_{k=1}^{n} \overline{z_k} = K$. Therefore, there exists an index $l > i$, such that $\overline{z_l} > 0$. We define $j = \operatorname{argmax}_{k\in\{1,\ldots,n\}} \overline{z_k} \neq 0$. We choose arbitrary $\varepsilon \in (0,\ \min\{\overline{z_j},\ 1 - \overline{z_i}\})$. We define $\hat{\mathbf{z}} = (\overline{z_1}, \ldots, \overline{z_{i-1}},\ \overline{z_i} + \varepsilon,\ \overline{z_{i+1}}, \ldots, \overline{z_{j-1}},\ \overline{z_j} - \varepsilon,\ \overline{z_{j+1}}, \ldots,\ \overline{z_n})$. Then

$$\sum_{k=1}^{n} \widehat{z_k} y_k = \sum_{k=1}^{n} \overline{z_k} y_k + \varepsilon \underbrace{(y_i - y_j)}_{\geq 0} \geq \sum_{k=1}^{n} \overline{z_k} y_k.$$

If $y_i > y_j$, then $\sum_{k=1}^{n} \widehat{z_k} y_k > \sum_{k=1}^{n} \overline{z_k} y_k$. Thus $\bar{\mathbf{z}} \neq \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \sum_{k=1}^{n} x_k y_k$. On the other hand, if $y_i = y_j$, then from the definition of $\mathbf{y}$ we obtain that

$$\forall k \in \{i, \ldots,\ j\},\ y_i = y_k = y_j. \tag{1}$$

We show that $j \geq \lfloor K \rfloor + 1$. Suppose to the contrary that $j \leq \lfloor K \rfloor$. $K = \sum_{k=1}^{j} \overline{z_k} \leq \sum_{k=1}^{j} 1 = j \leq \lfloor K \rfloor$. If $K \notin \mathbb{N}$, then $\lfloor K \rfloor < K$, thus we have a contradiction. If $K \in \mathbb{N}$, then $K = \lfloor K \rfloor = j$ and for all $k \in \{1,\ \ldots,\ j\}\ \overline{z_k} = 1$. Nevertheless,

$$z_k = \begin{cases} 1 & \text{if } k \in \{0, \ldots,\ \lfloor K \rfloor = j\}, \\ 0 & \text{if } k \in \{j+1, \ldots,\ n\}. \end{cases}$$

Thence $\bar{\mathbf{z}} = \mathbf{z}$, which contradicts the definition of $\bar{\mathbf{z}}$. Now we proof that if the eq. (1) holds, than $\sum_{k=1}^{n} z_k y_k =$

$\sum_{k=1}^{n} \overline{z_k} y_k$.

$$\begin{aligned}
\sum_{k=1}^{n} z_k y_k &= \sum_{k=1}^{i-1} z_k y_k + \sum_{k=i}^{j} z_k y_k + \sum_{k=j+1}^{\lfloor K \rfloor + 1} z_k y_k + \sum_{k=\lfloor K \rfloor + 2}^{n} z_k y_k \\
&= \sum_{k=1}^{i-1} z_k y_k + y_i \sum_{k=i}^{j} z_k \\
&= \sum_{k=1}^{i-1} \overline{z_k} y_k + y_i \sum_{k=i}^{j} \overline{z_k} \\
&= \sum_{k=1}^{n} \overline{z_k} y_k,
\end{aligned}$$

where the second equation follows from eq. (1), the fact that $j \geq \lfloor K \rfloor + 1$ and the definition of $\mathbf{z}$. The third equation holds because $K = \sum_{k=1}^{i-1} z_k + \sum_{k=i}^{n} z_k = \sum_{k=1}^{i-1} \overline{z_k} + \sum_{k=i}^{n} z_k$, thus $\sum_{k=i}^{n} z_k = \sum_{k=i}^{n} \overline{z_k}$. Therefore, we have a contradiction to the supposal that $\sum_{k=1}^{n} \overline{z_k} y_k > \sum_{k=1}^{n} z_k y_k$. Hence, the lemma holds. □

Next, we proceed with the proof of Lemma 4.9.

Proof of Lemma 4.9. Suppose to the contrary that $\forall S \subset C$ satisfying $|S| = 2^{\alpha(\lambda,k)-k+1} - 1$ it holds that $\sum_{c \in S} p_{\mathrm{ENC}}(m^*, c) < p(\lambda)$. We denote $K = \sum_{c \in C} p_{\mathrm{DEC}}(c, m^*) \leq 2^{\alpha(\lambda,k)-k+1} - 1$, where the last inequality comes from Lemma 4.10. That implies $K \leq |S|$.

The p-correctness can be expressed in the following ways:

$$\forall \lambda \in \mathbb{N},\, \forall m \in M,\, \Big(\Pr[\mathrm{Dec}(\mathrm{Enc}(1^\lambda, m)) = m] \Big) \geq p(\lambda),$$

$$\forall \lambda \in \mathbb{N}, \forall m \in M,\, \left(\sum_{c \in C} p_{\mathrm{ENC}}(m, c) p_{\mathrm{DEC}}(c, m) \right) \geq p(\lambda).$$

Next, we sort the C in a way that $\forall i, j \in \{1, \ldots, |C|\}$, $(i \leq j) \Rightarrow p_{\mathrm{ENC}}(m, c_i) \leq p_{\mathrm{ENC}}(m, c_j)$. Let $S = \{c_1, \ldots, c_{|S|}\}$, where $|S| = 2^{\alpha(\lambda,k)-k+1} - 1$. We assume that the distribution of $p_{\mathrm{DEC}}(m^*, c_i)$ is as follows

$$p_{\mathrm{DEC}}(m^*, c_i) = \begin{cases} 1 & \text{if } i \leq \lfloor K \rfloor, \\ K - \lfloor K \rfloor & \text{if } \lfloor K \rfloor < i \leq \lfloor K \rfloor + 1, \\ 0 & \text{if } i > \lfloor K \rfloor + 1. \end{cases}$$

The proof that the sum $\sum_{c \in C} p_{\mathrm{ENC}}(m^*, c) p_{\mathrm{DEC}}(c, m^*)$ is maximized for the p_{DEC} distribution defined above follows from Lemma 4.11, where $n = |C|$, $\mathbf{y} = p_{\mathrm{ENC}}(m^*, c)$, and $\mathbf{z} = p_{\mathrm{DEC}}(c, m^*)$. Then

$$\begin{aligned}
\sum_{c \in C} p_{\mathrm{ENC}}(m^*, c) p_{\mathrm{DEC}}(c, m^*) &= \sum_{i=1}^{|C|} p_{\mathrm{ENC}}(m^*, c_i) p_{\mathrm{DEC}}(c_i, m^*) \\
&= \sum_{i=1}^{|S|} p_{\mathrm{ENC}}(m^*, c_i) p_{\mathrm{DEC}}(c_i, m^*) \\
&\leq \sum_{c \in S} p_{\mathrm{ENC}}(m^*, c) \\
&< p(\lambda),
\end{aligned}$$

where the last inequality comes from our supposal. This contradicts the p-correctness definition, thus the proof is complete. □

Figure 5: Adversary $\mathcal{A}$

Adversary $\mathcal{A} = (\mathcal{A}.\text{Select}(1^\lambda), \mathcal{A}.\text{Compress}(\text{aux}, c), \mathcal{A}.\text{Expand}(\text{aux}, w))$

$\mathcal{A}.\text{Select}(1^\lambda)$:

- Choose $m^* = \text{argmin}_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*)$.
- Choose a bijection $g : S \to W$.
- Compute $g^{-1} : W \to S$ from g.
- Output $(m^*, \text{aux} = (g, g^{-1}))$.

$\mathcal{A}.\text{Compress}(\text{aux}, c)$:

- Parse aux as $\text{aux} = (g, g^{-1})$.
- Output $w = g(c)$

$\mathcal{A}.\text{Expand}(\text{aux}, w)$:

- Parse aux as $\text{aux} = (g, g^{-1})$.
- Output $c' = g^{-1}(w)$

We can now proceed with the proof of Theorem 4.1.

Proof of Theorem 4.1. Our construction of the adversary $\mathcal{A} = (\mathcal{A}.\text{Select}, \mathcal{A}.\text{Compress}, \mathcal{A}.\text{Expand})$ is given in Figure 5. The $\mathcal{A}.\text{Select}$ algorithm chooses a message m^* such that there exists a set $S \subset C$ such that every member of S can be uniquely represented using an index (bit string) with a length less than or equal to β bits and $\Pr[c \in S] = \sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$. Then the algorithm chooses a bijection g from S to the set of indices $W = \bigcup_{i=0}^{\alpha(\lambda,k)} \{0,1\}^i$. The $\mathcal{A}.\text{Compress}$ gets a code word c from the Enc algorithm and returns its index $w = f(c)$, if $c \in S$. If $c \notin S$, it returns an arbitrary element of W. Finally, the $\mathcal{A}.\text{Expand}$ returns $f^{-1}(w)$, which is equal to c if the c is in S.

The set S can be obtained by sorting the code words with respect to $p_{\text{ENC}}(m^*, c)$ and taking the first $2^{\alpha(\lambda,k)-k+1} - 1$ code words with the highest value $(p_{\text{ENC}}(m^*, c))$. The bijection g is possible to construct, because $|S| = 2^{\alpha(\lambda,k)-k+1} - 1 = |W|$.

Then, the $\text{CompExp}^{\Pi}_{\mathcal{A},\beta}$ follows:

- $\left(m^*, \text{aux} = (g, g^{-1})\right) \leftarrow \mathcal{A}.\text{Select}(1^\lambda)$
- $c \leftarrow \text{Enc}(1^\lambda, m^*)$
- $w \leftarrow \mathcal{A}.\text{Compress}(\text{aux}, c)$
- $c' = \mathcal{A}.\text{Expand}(\text{aux}, w)$
- if $(c' = c) \wedge |w| \leq \beta(\lambda, k)$ output 1, else output 0

The bound $|w| \leq \beta(\lambda, k)$ holds, because $w \in W$, $W = \bigcup_{i=0}^{\alpha(\lambda,k)-k} \{0,1\}^i$. The equality $c = c'$ holds if and only if $c \in S$. We know that $\Pr[c \in S] = \sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$ from the Lemma 4.9. Therefore the proof is complete. □

5 Conclusions

We proved that there cannot exist any non-trivial incompressible encoding schemes in the plain model secure against computationally unbounded adversaries. In the current version of the compression experiment in the definition of β-Incompressibility, the adversary $\mathcal{A}$ is allowed to choose the message to be encoded, and our impossibility results depend on it. We leave as an interesting open problem whether the results change if the message in the compression experiment is chosen randomly.

Acknowledgements

We wish to thank Daniel Wichs for clarifications and insightful comments on the presentation of this note.

References

[MW20] Tal Moran and Daniel Wichs. Incompressible encodings. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 494–523. Springer, 2020.